

08/18/00

8-21-00

A

PTO/SB/05 (4/98)

Approved for use through 09/30/2000. OMB 0651-0032

Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Please type a plus sign (+) inside this box → +

**UTILITY  
PATENT APPLICATION  
TRANSMITTAL**

(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))

Attorney Docket No. 103.1048.01

First Inventor or Application Identifier Scott SCHOENTHAL

Title Persistent and Reliable Delivery of Event Messages

Express Mail Label No. EL 524 781 075 US

**APPLICATION ELEMENTS**

See MPEP chapter 600 concerning utility patent application contents.

1. ☐ \* Fee Transmittal Form (e.g., PTO/SB/17)  
(Submit an original and a duplicate for fee processing)

2. ☒ Specification [Total Pages 45]  
(preferred arrangement set forth below)

- Descriptive title of the Invention
- Cross References to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings (if filed)
- Detailed Description
- Claim(s)
- Abstract of the Disclosure

3. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 4]

4. Oath or Declaration [Total Pages ]

- a. ☐ Newly executed (original or copy)
- b. ☐ Copy from a prior application (37 C.F.R. § 1.63(d))  
(for continuation/divisional with Box 16 completed)
- i. ☐ **DELETION OF INVENTOR(S)**  
Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).

**\* NOTE FOR ITEMS 1 & 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).**

5. ☐ Microfiche Computer Program (Appendix)

6. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)

- a. ☐ Computer Readable Copy
- b. ☐ Paper Copy (identical to computer copy)
- c. ☐ Statement verifying identity of above copies

**ACCOMPANYING APPLICATION PARTS**

7. ☐ Assignment Papers (cover sheet & document(s))

8. ☐ 37 C.F.R. § 3.73(b) Statement ☐ Power of Attorney  
(when there is an assignee)

9. ☐ English Translation Document (if applicable)

10. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations

11. ☐ Preliminary Amendment

12. ☒ Return Receipt Postcard (MPEP 503)  
(Should be specifically itemized)

13. ☐ \* Small Entity Statement(s) ☐ Statement filed in prior application, Status still proper and desired  
(PTO/SB/09-12)

14. ☐ Certified Copy of Priority Document(s)  
(if foreign priority is claimed)

15. ☒ Other: certificate of mailing

**16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:**

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: \_\_\_\_\_

Prior application information: Examiner \_\_\_\_\_ Group / Art Unit: \_\_\_\_\_

**For CONTINUATION or DIVISIONAL APPS only:** The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when information has been inadvertently omitted from the submitted application parts.

**17. CORRESPONDENCE ADDRESS**

☒ Customer Number or Bar Code Label 22883 (Insert Customer No. or Attach bar code label here)

or ☐ Correspondence address below

NAME: \_\_\_\_\_

Address: \_\_\_\_\_

City: \_\_\_\_\_ State: \_\_\_\_\_ Zip Code: \_\_\_\_\_

Country: \_\_\_\_\_ Telephone: \_\_\_\_\_ Fax: \_\_\_\_\_

Name (Print/Type) Steven A. Swernofsky Registration No. (Attorney/Agent) 33,040

Signature *Steven A. Swernofsky* Date August 18, 2000

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

**Certificate of Mailing (37 C.F.R. § 1.10)**

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Services on the date shown below as "Express Mail" (Post Office to Addressee) in an envelope addressed to the Honorable Assistant Commissioner for Patents and Trademarks, Box Patent Application, Washington, D.C. 20231.



Mailing Label No. EL 524 781 075 US

Date of Deposit: August 18, 2000

Arlette Malhas

Printed Name

*Arlette Malhas*  
Signature

Documents enclosed:

- Utility Patent Application Transmittal Form;
- Specification (26) pages;
- Claims (18) pages;
- Abstract (1) pages;
- Drawings (4) pages;
- Return post card; and
- Certificate of Express Mailing.

1 This application is submitted in the name of the following inventor(s):

2

3	<i>Inventor</i>	<i>Citizenship</i>	<i>Residence City and State</i>
4	Scott SCHOENTHAL	United States	San Ramon, California

5

6 The assignee is *Network Appliance, Inc.*, a California corporation having an  
7 office at 495 East Java Drive, Sunnyvale CA 94089.

8  
9 TITLE OF THE INVENTION

10  
11 Persistent and Reliable Delivery of Event Messages

12  
13 BACKGROUND OF THE INVENTION

14  
15 1. *Field of the Invention*

16  
17 This invention relates to persistent and reliable delivery of event messages,  
18 including event messages in file server systems in which it is desired to maintain reliable  
19 file system consistency.

20

1    2.    *Related Art*

2  
3            In systems that provide services to clients, such as those including file serv-  
4    ers and similar devices, it often occurs that the system, or some subsystem within that  
5    system, generates a message indicating the occurrence of a special event. Typically, the  
6    special event is an error of some kind, and the message conveys information regarding  
7    the nature of the special event, such as the type of error and the subsystem within which  
8    the error occurred. Many systems that provide services, including file servers, make ef-  
9    forts to assure that the services are reliably provided, and that the system providing the  
10   services is in a consistent state at all times. Thus, such systems find it advantageous to  
11   assure that all state information regarding the system, including state information relating  
12   to error messages, is persistently and reliably maintained. Such systems also find it ad-  
13   vantageous to assure that all event messages are reliably delivered and persistently main-  
14   tained until delivery is confirmed by the intended recipient of the event message.

15  
16            Accordingly, it would be advantageous to provide a technique for persistent  
17   and reliable delivery of event messages, that is not subject to the drawbacks of the known  
18   art. Preferably, those parts of the system responsible for delivering event messages are  
19   able to persistently maintain those event messages until delivery of those event messages  
20   has been confirmed by the intended recipient of the event message. Moreover, those  
21   parts of the system responsible for recovering from system crashes and other system er-

1 rors are able to persistently maintain those event messages until delivery, even after re-  
2 covery from system crashes or other system errors.

## 4 SUMMARY OF THE INVENTION

6 The invention provides a method and system for persistent and reliable de-  
7 livery of event messages, that is not subject to the drawbacks of the known art. Those  
8 parts of the system responsible for delivering event messages are able to persistently  
9 maintain those event messages until delivery of those event messages has been confirmed  
10 by the intended recipient of the event message. Those parts of the system responsible for  
11 recovering from system crashes and other system errors are able to persistently maintain  
12 those event messages until delivery, even after recovery from system crashes or other  
13 system errors.

15 In a first aspect of the invention, the system includes a set of event message  
16 producers, and maintains an event-indication queue of those event messages provided by  
17 the event producers using a set of pre-allocated resources. This first aspect allows the  
18 system to maintain the event-indication queue even when the event message indicates  
19 that allocation of new resources (such as for recording event messages) is unstable. An  
20 event-distribution engine distributes event messages to intended recipients and, after  
21 having received confirmation that the event messages were received, removes them from  
22 the event-indication queue. Recipients (also called “consumers”) of event messages re-

1 ceive the event messages, acknowledge their receipt thereof, and might take action in re-  
2 sponse to the event message.

3  
4 In a second aspect of the invention, the system includes a persistent mem-  
5 ory (such as NVRAM or other non-volatile memory), in which event messages can be re-  
6 corded until they are completely handled by the event-distribution engine. Upon recov-  
7 ery from a system crash or other system error, a replay-event producer retrieves those  
8 event messages recorded in the persistent memory and not yet completely handled, and  
9 re-presents them as event messages for the event-indication queue and the event-  
10 distribution engine.

11  
12 In a third aspect of the invention, the system includes an initialization  
13 memory (also called a "system boot memory"), in which event messages can be recorded  
14 until the system has completed its operations of initializing and becoming completely  
15 prepared to handle event messages. Upon recovery from a system crash or other system  
16 error, the replay-event producer retrieves those event messages recorded in the initializa-  
17 tion memory and re-presents them as event messages for the event-indication queue and  
18 the event-distribution engine.

19  
20 In a fourth aspect of the invention, a cluster of file servers collectively  
21 forming a highly-available system shares persistent memories. Each individual file  
22 server writes event messages to both its own and at least one other persistent memory, so

1 that upon a system crash or other system error, at least one other file server has a record  
2 of those event messages that were presented by event producers but not yet completely  
3 handled by the event-distribution engine. Upon indication of a system crash or other  
4 system error by a first file server in the cluster, a second file server in the cluster uses its  
5 replay-event producer. Thus, the second file server retrieves those event messages re-  
6 corded in the persistent memory and not yet completely handled by the first file server,  
7 and re-presents them as event messages for its own event-indication queue and event-  
8 distribution engine.

9  
10 In a fifth aspect of the invention, a cluster of event recipients can be cou-  
11 pled to a single multiplexing recipient that includes a second persistent memory in which  
12 event messages are recorded after receipt and before being redistributed or otherwise  
13 completely handled. Upon recovery from a system crash or other system error by the  
14 multiplexing recipient, the multiplexing recipient includes a replay-event producer that  
15 retrieves those event messages recorded in the second persistent memory, and re-presents  
16 them as event messages as if newly received from an event producer.

17  
18 The invention provides an enabling technology for a wide variety of appli-  
19 cations for persistent and reliable delivery of event messages, so as to obtain substantial  
20 advantages and capabilities that are novel and non-obvious in view of the known art. Ex-  
21 amples described below primarily relate to reliable file systems, but the invention is

1 broadly applicable to many different types of systems in which persistent and reliable de-  
2 livery of event messages is desired.

#### 4 BRIEF DESCRIPTION OF THE DRAWINGS

6 Figure 1 shows a block diagram of a portion of a system capable of persis-  
7 tent and reliable delivery of event messages.

9 Figure 2 shows a process flow diagram of a method for operating a system  
10 as in figure 1.

#### 12 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

14 In the following description, a preferred embodiment of the invention is de-  
15 scribed with regard to preferred process steps and data structures. Embodiments of the  
16 invention can be implemented using general-purpose processors or special purpose proc-  
17 essors operating under program control, or other circuits, adapted to particular process  
18 steps and data structures described herein. Implementation of the process steps and data  
19 structures described herein would not require undue experimentation or further invention.



*Related Applications*

Inventions described herein can be used in conjunction with technology described in the following documents.

- U.S. Patent Application Serial No. \_\_\_\_\_, Express Mail Mailing No. EL 524781089US, filed August 18, 2000, in the name of Blake LEWIS, attorney docket number 103.1033.01, titled “Reserving File System Blocks”
- U.S. Patent Application Serial No. \_\_\_\_\_, Express Mail Mailing No. EL524780242US, filed August 18, 2000, in the name of Rajesh SUNDARAM, attorney docket number 103.1034.01, titled “Dynamic Data Storage”
- U.S. Patent Application Serial No. \_\_\_\_\_, Express Mail Mailing No. EL524780239US, filed August 18, 2000, in the name of Blake LEWIS, attorney docket number 103.1035.01, titled “Instant Snapshot”
- U.S. Patent Application Serial No. \_\_\_\_\_, Express Mail Mailing No. EL524781092US, filed August 18, 2000, in the name of Douglas P. DOUCETTE, attorney docket number 103.1045.01, titled “Improved Space Allocation in a Write Anywhere File System”

1 and

- 2 • U.S. Patent Application Serial No. \_\_\_\_\_, Express Mail Mailing No.  
3 EL524780256US, filed August 18, 2000, in the name of Ray CHEN, attorney  
4 docket number 103.1047.01, titled “manipulation of Zombie Files and Evil-Twin  
5 Files”  
6

7 Each of these documents is hereby incorporated by reference as if fully set  
8 forth herein. This application claims priority of each of these documents. These docu-  
9 ments are collectively referred to as the “Incorporated Disclosures.”  
10

11 *Lexicography*  
12

13 The following terms refer or relate to aspects of the invention as described  
14 below. The descriptions of general meanings of these terms are not intended to be limit-  
15 ing, only illustrative.  
16

- 17 • *event messages* --- In general, an event message refers to an alert or notification of  
18 a system event, for which an intended recipient of that event message may wish to  
19 receive and possibly take action in response to. Examples of event messages in-  
20 clude notification of errors, or of events to be logged or otherwise administratively  
21 monitored.  
22

- 1 • *event replay* --- In general, re-presenting event messages (from a memory or other  
2 record) by a system element other than the one that generated the original event  
3 message. Event messages that are replayed are treated substantially identically to  
4 originally generated event messages.  
5
- 6 • *persistent, reliable* --- In general, persistent refers to memory or another record  
7 that is capable of surviving a disruption such as a system crash or a system error.  
8 In general, reliable refers to a process that is capable of being performed atomi-  
9 cally or otherwise completely even in the event of a disruption such as a system  
10 crash or a system error.  
11
- 12 • *pre-allocated resources* --- In general, resources that have been allocated ahead of  
13 time for use by a system element (such as for delivery of event messages), so that  
14 system element can operate even when disruptions in system operation make it  
15 uncertain that resource allocation will be effective at the time when the system  
16 element needs those resources.  
17
- 18 • *system crashes, system errors* --- In general, a system crash or a system error re-  
19 fers to a disruption in system operation sufficiently serious as to place the con-  
20 tinuation of system operations (such as delivery of event messages) in doubt. In  
21 the description herein, it is assumed that continuation of system operations does  
22 not generally survive such disruptions.

- 1
- 2 • *system initialization* --- In general, a time during which a system (such as a file  
3 server) is booting or initializing, and so is not necessarily able to perform all nec-  
4 essary operations (such as allocation of resources or delivery of event messages).
- 5

6 As noted above, these descriptions of general meanings of these terms are  
7 not intended to be limiting, only illustrative. Other and further applications of the inven-  
8 tion, including extensions of these terms and concepts, would be clear to those of ordi-  
9 nary skill in the art after perusing this application. These other and further applications  
10 are part of the scope and spirit of the invention, and would be clear to those of ordinary  
11 skill in the art, without further invention or undue experimentation.

12

### 13 *System Elements*

14

15 Figure 1 shows a block diagram of a portion of a system capable of persis-  
16 tent and reliable delivery of event messages.

17

18 A system 100 includes a set of event producers 110, a set of pre-allocated  
19 initialization event message resources 120, a set of pre-allocated post-initialization event  
20 message resources 130, a persistent memory 140, an event indication queue 150, an event  
21 distribution engine 160, a set of event recipients 170 including at least one multiplexing

1 recipient 171 and a set of intended recipients 172, a second persistent memory 180 at the  
2 multiplexing recipient 171, and an event replay engine 190.

3  
4 The event producers 110 include system elements, such as software mod-  
5 ules or hardware circuits, each capable of generating at least one event message 111 for  
6 delivery to at least one intended recipient 172. In a preferred embodiment, each event  
7 message 111 has a standardized format, including information about the time the event  
8 was recognized, the system element that recognized the event, the nature of the event,  
9 and any detailed information about the event necessary or desirable for the intended re-  
10 cipient 172 to know.

11  
12 The pre-allocated initialization event message resources 120 include mem-  
13 ory, and possibly other resources, for recording and maintaining event messages for fur-  
14 ther processing. In a preferred embodiment, the pre-allocated initialization event mes-  
15 sage resources 120 include resources allocated by the system 100 prior to generation of  
16 any new event messages 111. In a preferred embodiment, the event messages 111 are re-  
17 corded in the pre-allocated initialization event message resources 120 before being proc-  
18 essed by the event distribution engine 160.

19  
20 The pre-allocated post-initialization event message resources 130, similar  
21 to the pre-allocated initialization event message resources 120, memory, and possibly  
22 other resources, for recording and maintaining event messages for further processing. In

1 a preferred embodiment, the pre-allocated post-initialization event message resources 130  
2 include resources allocated by the system 100 during an initialization period and prior to  
3 generation of any new event messages 111. In a preferred embodiment, event messages  
4 111 that are generated during the system initialization period are recorded in the pre-  
5 allocated post-initialization event message resources 130. After the system initialization  
6 period, event messages 111 recorded therein are processed by the event distribution en-  
7 gine 160.

8  
9 The persistent memory 140 includes a memory, such as NVRAM, SRAM,  
10 or other memory whose contents are expected to survive a system crash or system error.  
11 In a preferred embodiment, the persistent memory 140 includes NVRAM also used with  
12 the WAFL file system. However, in alternative embodiments, the persistent memory 140  
13 may include any other form of persistent memory, whether NVRAM or not, and whether  
14 coordinating with aspects of the WAFL file system or not. Thus, upon recovery from a  
15 system crash or system error, the persistent memory 140 will still record those event mes-  
16 sages 111 that were not fully processed before the system crash or system error.

17  
18 The event indication queue 150 includes a memory having a queue of in-  
19 formation about event messages 111 (such as the event messages 111 themselves).

20  
21 The event distribution engine 160 includes a system element capable of  
22 reading information about event messages 111 from the event indication queue 150 and

1 capable of delivering at those event messages 111 to intended recipients 172 thereof. In a  
2 preferred embodiment, the event distribution engine 160 includes a software module in  
3 the system 100.

4  
5 The event recipients 170 (including at least one multiplexing recipient 171  
6 and a set of intended recipients 172) include system elements, possibly at remote devices  
7 such as clients for the file server system 100, capable of receiving event messages 111  
8 and deciding whether or not to act in response to those event messages 111. In a pre-  
9 ferred embodiment, actions take with regard to event messages 111 can include alerts or  
10 notification of selected users (such as a system operator), logging the event messages  
11 111, or maintaining statistics with regard thereto.

12  
13 The second persistent memory 180 at the multiplexing recipient 171 in-  
14 cludes, similar to the persistent memory 140, a memory, such as NVRAM, SRAM, or  
15 other memory whose contents are expected to survive a system crash or system error.  
16 The multiplexing recipient 171 includes a recipient replay element 181, capable of read-  
17 ing information about event messages 111 from the second persistent memory 180 and  
18 capable of replaying those event messages 111 as if newly received by the multiplexing  
19 recipient 171 (thus delivering those event messages 111 to the intended recipients 172).

20  
21 The event replay engine 190 includes a system element capable of reading  
22 information about event messages 111 from the persistent memory 140 and capable of

1 replaying those event messages 111 as if newly generated. The replay element 190 in-  
2 cludes a system initialization replay sub-element 191, an incomplete event distribution  
3 replay sub-element 192, and a cooperating systems replay sub-element 193.

#### 5 *Method of Operation*

7 Figure 2 shows a process flow diagram of a method for operating a system  
8 as in figure 1.

10 A method 200 includes a set of flow points and a set of steps. The system  
11 100 performs the method 200. Although the method 200 is described serially, the steps of  
12 the method 200 can be performed by separate elements in conjunction or in parallel,  
13 whether asynchronously, in a pipelined manner, or otherwise. There is no particular re-  
14 quirement that the method 200 be performed in the same order in which this description  
15 lists the steps, except where so indicated.

17 As described below, the method 200 includes a set of processes, each of  
18 which has a set of tasks operating independently and asynchronously with regard to each  
19 other.

#### 21 *1. Processing Event Messages*



1 A first process in the method 200 is described with regard to a flow point  
2 210, a flow point 220, and steps there-between. This first process includes a set of three  
3 tasks, each of which operates independently and asynchronously with regard to each  
4 other.

5  
6 At the flow point 210, the system 100 is ready to receive an event message  
7 111.

8  
9 *Event Generation*  
10

11 A first task includes a sequence including a step 211, a step 212, a step 213,  
12 and a step 214. In a preferred embodiment, the first task in its process includes these  
13 steps being performed in sequence and repeatedly.  
14

15 At the step 211, an event producer 110 generates an event message 111.  
16

17 At the step 212, if the system 100 is in normal operation, the event message  
18 111 is recorded in the pre-allocated initialization event message resources 120. If the  
19 system 100 is still in its initialization time duration, the event message 111 is recorded in  
20 the pre-allocated post-initialization event message resources 130.  
21

1           At the step 213, the system 100 copies information about the event message  
2 111 to a set of locations in the persistent memory 140. In a preferred embodiment, the  
3 persistent memory 140 includes a set of memory sections 141, each persistently main-  
4 taining system information; at least one of these memory sections 141 maintains infor-  
5 mation about event messages 111. Other memory sections 141 maintain information  
6 about other aspects of the system 100, such as a consistency state of the file system or a  
7 set of incomplete file system requests.

8  
9           In a preferred embodiment, the memory sections 141 associated with event  
10 messages 111 maintain information about those event messages 111 in a FIFO having a  
11 head pointer and a tail pointer. FIFOs are known in the art of computer data storage.  
12 When information about a new event message 111 is recorded in the persistent memory  
13 140, the FIFO is updated to add the information about the new event message 111 to an  
14 end of the list. When confirmation is received that the event message 111 was delivered  
15 to its intended recipients 172, the FIFO is updated to remove the information about the  
16 event message 111.

17  
18           At the step 214, similar to the step 213, the system 100 copies information  
19 about the event message 111 to the event indication queue 150. In a preferred embodi-  
20 ment, the event indication queue 150 includes a FIFO similar to that maintained in the  
21 persistent memory 140.

1                   *Event Distribution*

2  
3                   A second task includes a sequence including a step 215. In a preferred em-  
4 bodiment, the second task in its process includes this step being performed repeatedly.

5  
6                   At the step 215, the event distribution engine 160 responds to the informa-  
7 tion about the event message 111 in the event indication queue 150. The event distribu-  
8 tion engine 160 delivers the event message 111 to its intended recipients 172. As part of  
9 this step, each particular intended recipient 172, when it receives the event message 111,  
10 responds to the event distribution engine 160 to confirm its receipt of the event message  
11 111.

12  
13                   *Event Confirmation*

14  
15                   A third task includes a sequence including a step 216 and a step 217. In a  
16 preferred embodiment, the third task in its process includes these steps being performed  
17 in sequence and repeatedly.

18  
19                   At the step 216, the event distribution engine 160 awaits confirmation from  
20 each intended recipient 172 that the event message 111 was received by that particular  
21 intended recipient 172. When the event distribution engine 160 receives confirmation  
22 from all intended recipients 172, the method proceeds with the next step.

At the step 217, the event distribution engine 160 removes the information about the event message 111 from the event indication queue 150 and from the persistent memory 140.

At the flow point 220, the system 100 has completely processed the event message 111.

## 2. *Replaying Event Messages*

At a flow point 230, the system 100 has recovered from a system crash or a system error.

At a step 231, the event replay engine 190 reads information about event messages 111 from the persistent memory 140. As part of this step, the event replay engine 190 performs three sub-steps 231(a), 231(b), and 231(c).

At the sub-step 231(a), the system initialization replay sub-element 191 reads information about event messages 111 associated with the pre-allocated post-initialization event message resources 130. The event replay engine 190 replays these event messages 111.

1           At the sub-step 231(b), the incomplete event distribution replay sub-  
2 element 192 reads information about event messages 111 associated with the pre-  
3 allocated initialization event message resources 120. The event replay engine 190 re-  
4 plays these event messages 111 next.

5  
6           At the sub-step 231(c), the cooperating systems replay sub-element 193  
7 reads information about event messages 111, from the persistent memory 140, associated  
8 with and stored there by a cooperating system 100. The event replay engine 190 replays  
9 these event messages 111 only if the cooperating system 100 is not operational at the  
10 time.

11  
12           In a preferred embodiment, multiple cooperating systems 100 (preferably a  
13 pair of exactly two) are each capable of reading and writing to each other's persistent  
14 memories 140. Thus, when a first cooperating system 100 in the pair writes to its persis-  
15 tent memory 140, the second cooperating system 100 in the pair is able to read from that  
16 persistent memory 140. If the first cooperating system 100 suffers a system crash or  
17 system error, the second cooperating system 100, upon recognizing that system crash or  
18 system error, proceeds to replay the event messages 111 from the first cooperating sys-  
19 tem's persistent memory 140. Operation of multiple cooperating systems 100 is further  
20 described in the Incorporated Disclosures, particularly with regard to techniques used to  
21 prevent multiple cooperating systems 100 from disrupting each other's operation.

As noted herein, “replay” of event messages 111 is treated by the event indication queue 150 and the event distribution engine 160 as if the event messages 111 were newly generated. Replayed event messages 111 are processed and delivered before new event messages 111, according to the portion of the method 200 described with regard to flow point 210 and flow point 220.

At a flow point 240, the system 100 has replayed all event messages 111 not yet fully processed, and is ready to proceed at the flow point 210.

### 3. *Multiplexing Recipient Operation*

A third process in the method 200 is described with regard to a flow point 250, a flow point 260, and steps there-between. Similar to the first process, this third process includes a set of three tasks, each of which operates independently and asynchronously with regard to each other.

At the flow point 250, a multiplexing recipient 171 is ready to receive an event message 111.

### *Event Reception*

1           A first task includes a sequence including a step 251, a step 252, and a step  
2   253. In a preferred embodiment, this first task in its process includes these steps being  
3   performed in sequence and repeatedly.

4  
5           At the step 251, the multiplexing recipient 171 receives the event message  
6   111 from the event distribution engine 160.

7  
8           At the step 252, similarly to the steps described with regard to the flow  
9   point 210 and the flow point 220, the multiplexing recipient 171 records information  
10   about the event message 111 in its second persistent memory 180.

11  
12           At the step 253, the multiplexing recipient 171 (optionally) responds to the  
13   event message 111 by confirming that it was received at the multiplexing recipient 171  
14   (but not necessarily at the intended recipients 172).

15  
16           *Event Multiplexing*

17  
18           A second task includes a sequence including a step 254. In a preferred em-  
19   bodiment, this second task in its process includes this step being performed repeatedly.

20  
21           At the step 254, the multiplexing recipient 171 (optionally) determines to  
22   which intended recipients 172 to deliver the event message 111. In a preferred embodi-

ment, the multiplexing recipient 171 filters the event messages 111 it receives, so that it delivers only those event messages 111 it receives to their actual intended recipients 172. For example, a particular intended recipient 172 might determine that it is only interested in a particular subclass of event messages 111. In such cases, the multiplexing recipient 171 delivers only that particular subclass of event messages 111 to that particular intended recipient 172.

### *Event Confirmation*

A third task includes a sequence including a step 255 and a step 256. In a preferred embodiment, this third task in its process includes these steps being performed in sequence and repeatedly.

At the step 255, the multiplexing recipient 171 awaits confirmation from each particular intended recipient 172 that the particular intended recipient 172 has received the event message 111.

At the step 256, the multiplexing recipient 171 receives such confirmation from individual intended recipients 172. As part of this step, the multiplexing recipient 171 (optionally) forwards those confirmations on to the event distribution engine 160. When the multiplexing recipient 171 receives all such confirmations, it removes the information about the event message 111 from the second persistent memory 180.



At the flow point 260, the multiplexing recipient 171 has completely processed the event message 111.

#### 4. *Replaying Multiplexed Event Messages*

At a flow point 270, the multiplexing recipient 171 has recovered from a system crash or a system error.

At a step 271, the multiplexing recipient 171 reads information about event messages 111 from the second persistent memory 180.

At a step 272, the multiplexing recipient 171 replays the event messages 111 from the second persistent memory 180.

“Replay” of event messages 111 by the multiplexing recipient 171 is similar to replay of event messages as described above with regard to the event indication queue 150 and the event distribution engine 160.

At a flow point 280, the multiplexing recipient 171 has replayed all event messages 111 not yet fully processed, and is ready to proceed at the flow point 250.

1     5.     *Confirming Event Messages*

2  
3             As described above, there are steps at which the system 100 or the multi-  
4     plexing recipient 171 awaits confirmation of the event message 111 from the intended re-  
5     cipient 172. In a preferred embodiment, confirmation of event messages 111 is per-  
6     formed by each intended recipient 172 as described below with regard to a flow point  
7     290, a flow point 300, and steps there-between.

8  
9             At the flow point 290, the intended recipient 172 is ready to receive an  
10    event message 111.

11  
12            At a step 291, the intended recipient 172 receives an event message 111.

13  
14            At a step 292, the intended recipient 172 parses the event message 111 and  
15    processes the event message 111 according to its own (internal) processing rules for that  
16    event message 111.

17  
18            At a step 293, the intended recipient 172 generates a confirmation message  
19    and sends that confirmation message to the sender of the event message 111.

20  
21            At the flow point 300, the intended recipient 172 has received, processed,  
22    and confirmed the event message 111. The sender of the event message 111, upon re-

1 ceipt of the confirmation message, can regard the event message 111 as completely han-  
2 dled and can safely delete it.

3  
4 *Generality of the Invention*  
5

6 The invention has general applicability to various fields of use, not neces-  
7 sarily related to the services described above. For example, these fields of use can include  
8 one or more of, or some combination of, the following:

- 9
- 10 • The invention is applicable to persistent and reliable delivery of messages other  
11 than event messages.
  - 12
  - 13 • The invention is applicable to persistent and reliable operation of other processes  
14 than delivery of messages.
  - 15
  - 16 • The invention is applicable to mutually cooperating systems to perform other per-  
17 sistent and reliable operations.
  - 18
  - 19 • The invention is applicable to hierarchical cooperating systems to perform other  
20 persistent and reliable operations.
  - 21



## CLAIMS

1  
2  
3 1. A method including

4 persistently maintaining at least one event message until at least one in-  
5 tended recipient of said event message confirms delivery of said event message; and

6 upon recovery from an error, replaying said event message;

7 whereby said event message is reliably delivered to said intended recipient.

8  
9 2. A method as in claim 1, including

10 receiving said event message by said intended recipient; and

11 generating a confirmation of said event message in response to said event  
12 message.

13  
14 3. A method as in claim 1, wherein said event message is provided by

15 at least one event message producer.

16  
17 4. A method as in claim 1, wherein said persistent maintenance in-

18 cludes recording said event message in an event-indication queue, said event-indication

19 queue having resources pre-allocated before occurrence of an event associated with said  
20 event message.

1           5.     A method as in claim 1, wherein said persistent maintenance in-  
2 cludes recording said event message in an event-indication queue, wherein said event-  
3 indication queue is reliable even when the event message indicates that allocation of new  
4 resources is unstable.

5  
6           6.     A method as in claim 1, wherein said persistent maintenance in-  
7 cludes recording said event message during a duration when delivery of said event mes-  
8 sage is not yet feasible.

9  
10           7.     A method as in claim 6, including  
11 upon termination of said duration, replaying said event message;  
12 whereby said event message is reliably delivered to said intended recipient.

13  
14           8.     A method as in claim 6, wherein said duration includes a boot time  
15 or an initialization time.

16  
17           9.     A method as in claim 1, wherein said persistent maintenance in-  
18 cludes recording said event message in a persistent memory.

19  
20           10.    A method as in claim 9, including  
21 delivering said event message to said intended recipient;  
22 receiving a confirmation of said delivery; and

1 removing said event message from said persistent memory in response to  
2 said confirmation.

3  
4 11. A method including  
5 persistently maintaining at least one event message during a duration when  
6 delivery of said event message is not yet feasible; and  
7 upon termination of said duration, replaying said event message;  
8 whereby said event message is reliably delivered to an intended recipient of  
9 said event message.

10  
11 12. A method as in claim 11, wherein said duration includes a boot time  
12 or an initialization time.

13  
14 13. A method as in claim 11, wherein said event message is provided by  
15 at least one event message producer.

16  
17 14. A method as in claim 11, including persistently maintaining at least  
18 one event message until at least one intended recipient of said event message confirms  
19 delivery of said event message.

20  
21 15. A method as in claim 14, including  
22 upon recovery from an error, replaying said event message;

1           whereby said event message is reliably delivered to said intended recipient.

2  
3           16.    A method as in claim 14, wherein said persistent maintenance in-  
4 cludes recording said event message in an event-indication queue, said event-indication  
5 queue having resources pre-allocated before occurrence of an event associated with said  
6 event message.

7  
8           17.    A method as in claim 14, wherein said persistent maintenance in-  
9 cludes recording said event message in an event-indication queue, wherein said event-  
10 indication queue is reliable even when the event message indicates that allocation of new  
11 resources is unstable.

12  
13           18.    A method as in claim 11, wherein said persistent maintenance in-  
14 cludes recording said event message in a persistent memory.

15  
16           19.    A method as in claim 18, including  
17 delivering said event message to said intended recipient;  
18 receiving a confirmation of said delivery; and  
19 removing said event message from said persistent memory in response to  
20 said confirmation.

21  
22           20.    A method as in claim 11, including



1 receiving said event message by said intended recipient; and  
2 generating a confirmation of said event message in response to said event  
3 message.

4  
5 21. A method including  
6 maintaining at least one event message in a plurality of memory locations,  
7 each one of said plurality of memory locations being accessible by both a first server de-  
8 vice and a second server device; and  
9 upon recovery from an error at said first server device, replaying said event  
10 message from said second server device;  
11 whereby said event message is reliably delivered to an intended recipient of  
12 said event message.

13  
14 22. A method as in claim 21, wherein said event message is provided by  
15 at least one event message producer.

16  
17 23. A method as in claim 21, wherein said maintenance includes persis-  
18 tently maintaining said event message during a duration when delivery of said event mes-  
19 sage is not yet feasible.

20  
21 24. A method as in claim 23, including  
22 upon termination of said duration, replaying said event message;

1           whereby said event message is reliably delivered to an intended recipient of  
2 said event message.

3  
4           25.    A method as in claim 23, wherein said duration includes a boot time  
5 or an initialization time.

6  
7           26.    A method as in claim 23, wherein said event message is provided by  
8 at least one event message producer.

9  
10          27.    A method as in claim 21, wherein said maintenance includes persis-  
11 tently maintaining said event message until at least one intended recipient of said event  
12 message confirms delivery thereof.

13  
14          28.    A method as in claim 27, wherein said persistent maintenance in-  
15 cludes recording said event message in an event-indication queue, said event-indication  
16 queue having resources pre-allocated before occurrence of an event associated with said  
17 event message.

18  
19          29.    A method as in claim 27, wherein said persistent maintenance in-  
20 cludes recording said event message in an event-indication queue, wherein said event-  
21 indication queue is reliable even when the event message indicates that allocation of new  
22 resources is unstable.

1  
2           30.   A method as in claim 27, wherein said persistent maintenance in-  
3 cludes recording said event message in a persistent memory.  
4

5           31.   A method as in claim 30, including  
6           delivering said event message to said intended recipient;  
7           receiving a confirmation of said delivery; and  
8           removing said event message from said persistent memory in response to  
9 said confirmation.  
10

11           32.   A method as in claim 30, including  
12           receiving said event message by said intended recipient; and  
13           generating a confirmation of said event message in response to said event  
14 message.  
15

16           33.   A method including  
17           delivering at least one event message to a multiplexing recipient;  
18           maintaining said event message at said multiplexing recipient; and  
19           reliably delivering said event message from said multiplexing recipient to at  
20 least one intended recipient of said event message.  
21

22           34.   A method as in claim 33, including

1 receiving said event message by said intended recipient; and  
2 generating a confirmation of said event message in response to said event  
3 message.

4  
5 35. A method as in claim 33, wherein said event message is provided by  
6 at least one event message producer.

7  
8 36. A method as in claim 33, wherein reliable delivery of said event  
9 message from said multiplexing recipient includes

10 persistently maintaining said event message at said multiplexing recipient;  
11 upon recovery from an error at said multiplexing recipient, replaying said  
12 event message from said multiplexing recipient to said intended recipient;

13 whereby said event message is reliably delivered to said intended recipient.

14  
15 37. A method as in claim 36, wherein said persistent maintenance in-  
16 cludes recording said event message in an event-indication queue, said event-indication  
17 queue having resources pre-allocated before occurrence of an event associated with said  
18 event message.

19  
20 38. A method as in claim 36, wherein said persistent maintenance in-  
21 cludes recording said event message in an event-indication queue, wherein said event-

1 indication queue is reliable even when the event message indicates that allocation of new  
2 resources is unstable.

3  
4 39. A method as in claim 36, wherein said persistent maintenance in-  
5 cludes recording said event message in a persistent memory.

6  
7 40. A method as in claim 39, including  
8 delivering said event message to said intended recipient;  
9 receiving a confirmation of said delivery; and  
10 removing said event message from said persistent memory in response to  
11 said confirmation.

12  
13 41. A method as in claim 33, wherein reliable delivery of said event  
14 message from said multiplexing recipient includes  
15 persistently maintaining said event message at said multiplexing recipient  
16 until at least one intended recipient of said event message confirms delivery of said event  
17 message;  
18 sending a confirmation of delivery from said multiplexing recipient in re-  
19 sponse to a confirmation of delivery from said intended recipient.

20  
21 42. A method as in claim 41, wherein said persistent maintenance in-  
22 cludes recording said event message in an event-indication queue, said event-indication

1 queue having resources pre-allocated before occurrence of an event associated with said  
2 event message.

3  
4 43. A method as in claim 41, wherein said persistent maintenance in-  
5 cludes recording said event message in an event-indication queue, wherein said event-  
6 indication queue is reliable even when the event message indicates that allocation of new  
7 resources is unstable.

8  
9 44. A method as in claim 36, wherein said persistent maintenance in-  
10 cludes recording said event message in a persistent memory.

11  
12 45. A method as in claim 44, including  
13 delivering said event message to said intended recipient;  
14 receiving a confirmation of said delivery; and  
15 removing said event message from said persistent memory in response to  
16 said confirmation.

17  
18 46. A memory including instructions, said instructions capable of being  
19 interpreted to indicate

20 persistently maintaining at least one event message until at least one in-  
21 tended recipient of said event message confirms delivery of said event message; and

22 upon recovery from an error, replaying said event message;

1           whereby said event message is reliably delivered to said intended recipient.

2  
3           47.    A memory as in claim 46, wherein said instructions are also capable  
4   of being interpreted to indicate recording said event message during a duration when de-  
5   livery of said event message is not yet feasible.

6  
7           48.    A memory including instructions, said instructions capable of being  
8   interpreted to indicate

9           maintaining at least one event message in a plurality of memory locations,  
10   each one of said plurality of memory locations being accessible by both a first server de-  
11   vice and a second server device; and

12           upon recovery from an error at said first server device, replaying said event  
13   message from said second server device;

14           whereby said event message is reliably delivered to an intended recipient of  
15   said event message.

16  
17           49.    A memory including instructions, said instructions capable of being  
18   interpreted to indicate

19           delivering at least one event message to a multiplexing recipient;

20           maintaining said event message at said multiplexing recipient; and

21           reliably delivering said event message from said multiplexing recipient to at  
22   least one intended recipient of said event message.

1  
2           50.   Apparatus including

3               means for persistently maintaining at least one event message until at least  
4 one intended recipient of said event message confirms delivery of said event message;  
5 and

6               means for replaying said event message upon recovery from an error.  
7

8           51.   Apparatus as in claim 50, including

9               means for receiving said event message by said intended recipient; and  
10              means for generating a confirmation of said event message in response to  
11 said event message.  
12

13           52.   Apparatus as in claim 50, wherein said means for persistently main-  
14 taining includes means for recording said event message in an event-indication queue,  
15 said event-indication queue having resources pre-allocated before occurrence of an event  
16 associated with said event message.  
17

18           53.   Apparatus as in claim 50, wherein said means for persistently main-  
19 taining includes means for recording said event message in an event-indication queue,  
20 wherein said event-indication queue is reliable even when the event message indicates  
21 that allocation of new resources is unstable.  
22



1           54. Apparatus as in claim 50, wherein said means for persistently main-  
2   taining includes means for recording said event message during a duration when delivery  
3   of said event message is not yet feasible.

4  
5           55. Apparatus as in claim 54, including  
6           upon termination of said duration, means for replaying said event message;  
7           whereby said event message is reliably delivered to said intended recipient.

8  
9           56. Apparatus as in claim 54, wherein said duration includes a boot time  
10   or an initialization time.

11  
12           57. Apparatus as in claim 50, wherein said means for persistently main-  
13   taining includes means for recording said event message in a persistent memory.

14  
15           58. Apparatus as in claim 57, including  
16           means for delivering said event message to said intended recipient;  
17           means for receiving a confirmation of said delivery; and  
18           means for removing said event message from said persistent memory in re-  
19   sponse to said confirmation.

20  
21           59. Apparatus including

1 means for persistently maintaining at least one event message during a du-  
2 ration when delivery of said event message is not yet feasible; and  
3 upon termination of said duration, means for replaying said event message.  
4

5 60. Apparatus as in claim 59, wherein said duration includes a boot time  
6 or an initialization time.  
7

8 61. Apparatus as in claim 59, including means for persistently main-  
9 taining at least one event message until at least one intended recipient of said event mes-  
10 sage confirms delivery of said event message.  
11

12 62. Apparatus as in claim 61, including, upon recovery from an error,  
13 means for replaying said event message.  
14

15 63. Apparatus as in claim 61, wherein said means for persistently main-  
16 taining includes means for recording said event message in an event-indication queue,  
17 said event-indication queue having resources pre-allocated before occurrence of an event  
18 associated with said event message.  
19

20 64. Apparatus as in claim 61, wherein said means for persistently main-  
21 taining includes means for recording said event message in an event-indication queue,

1 wherein said event-indication queue is reliable even when the event message indicates  
2 that allocation of new resources is unstable.

3  
4 65. Apparatus as in claim 59, wherein said means for persistently main-  
5 taining includes means for recording said event message in a persistent memory.

6  
7 66. Apparatus as in claim 65, including  
8 means for delivering said event message to said intended recipient;  
9 means for receiving a confirmation of said delivery; and  
10 means for removing said event message from said persistent memory in re-  
11 sponse to said confirmation.

12  
13 67. Apparatus as in claim 65, including  
14 means for receiving said event message by said intended recipient; and  
15 means for generating a confirmation of said event message in response to  
16 said event message.

17  
18 68. Apparatus including  
19 means for maintaining at least one event message in a plurality of memory  
20 locations, each one of said plurality of memory locations being accessible by both a first  
21 server device and a second server device; and

1           upon recovery from an error at said first server device, means for replaying  
2   said event message from said second server device.

3  
4           69.   Apparatus including  
5           means for delivering at least one event message to a multiplexing recipient;  
6           means for maintaining said event message at said multiplexing recipient;  
7   and

8           means for reliably delivering said event message from said multiplexing re-  
9   cipient to at least one intended recipient of said event message.

10  
11           70.   Apparatus as in claim 69, including  
12           means for receiving said event message by said intended recipient; and  
13           means for generating a confirmation of said event message in response to  
14   said event message.

15  
16           71.   In a method including reliable delivery of event messages, a memory  
17   including  
18           a persistent record of at least one event message until at least one intended  
19   recipient of said event message confirms delivery of said event message; and  
20           upon recovery from an error, a replayable instance of said event message.

1           72.    A memory as in claim 71, including a record of said event message  
2 during a duration when delivery of said event message is not yet feasible.

3  
4           73.    A memory as in claim 71, including  
5 at least one event message in a plurality of memory locations, each one of  
6 said plurality of memory locations being accessible by both a first server device and a  
7 second server device; and

8 upon recovery from an error at said first server device, at least one instance  
9 of said event message replayable from said second server device.

10  
11           74.    In a method including reliable delivery of event messages, a memory  
12 including

13 a persistent record of at least one event message at a multiplexing recipient;  
14 and

15 an instance of said event message deliverable from said multiplexing re-  
16 cipient to at least one intended recipient of said event message.

17  
18           75.    In apparatus having elements capable of performing a method, said  
19 method including reliable delivery of event messages, a memory including

20 a persistent record of at least one event message until at least one intended  
21 recipient of said event message confirms delivery of said event message; and

22 upon recovery from an error, a replayable instance of said event message.

1

2           76.    A memory as in claim 75, including a record of said event message  
3 during a duration when delivery of said event message is not yet feasible.

4

5           77.    A memory as in claim 75, including  
6           at least one event message in a plurality of memory locations, each one of  
7 said plurality of memory locations being accessible by both a first server device and a  
8 second server device; and

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

upon recovery from an error at said first server device, at least one instance  
of said event message replayable from said second server device.

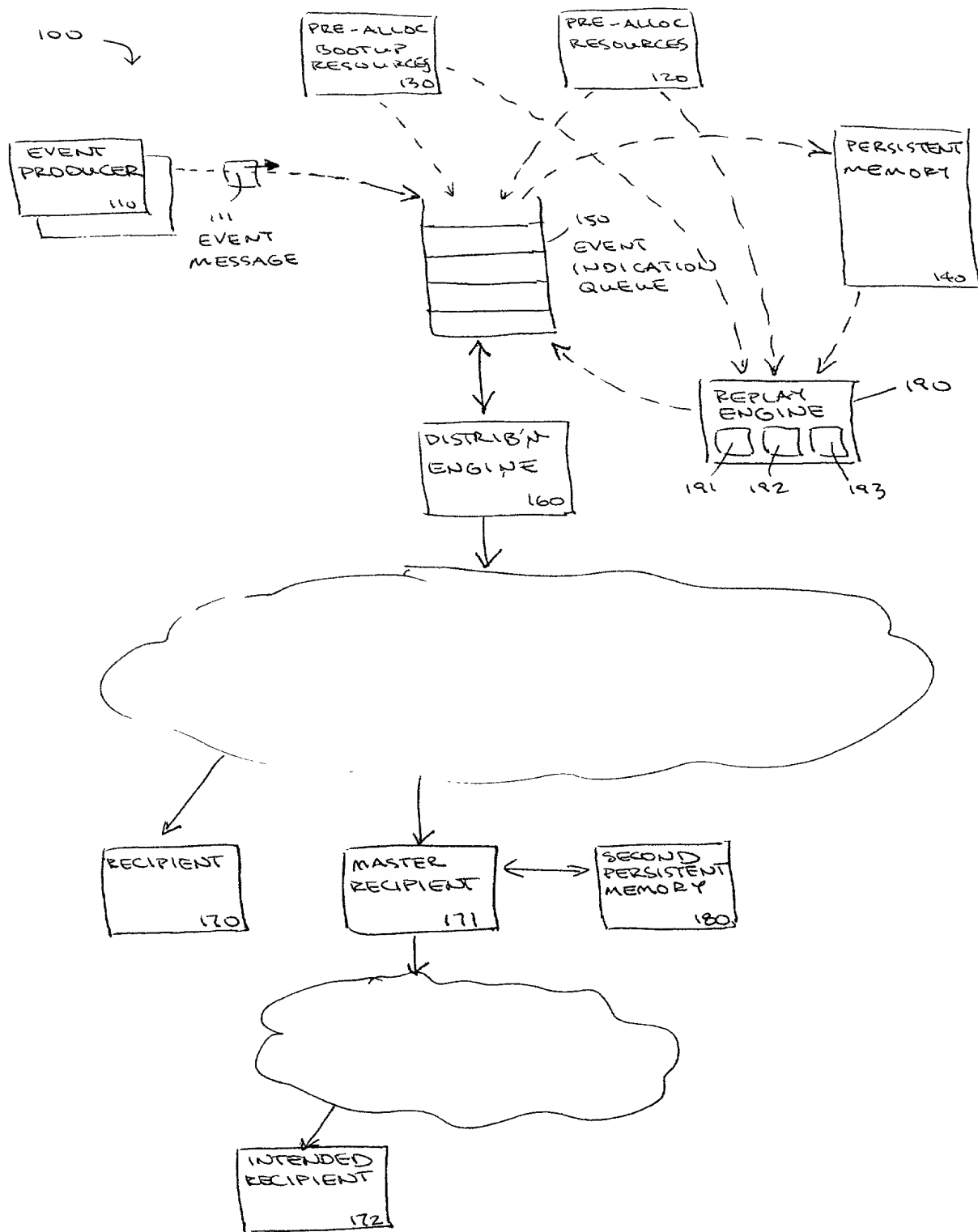
78.    A memory as in claim 75, including  
a persistent record of at least one event message at a multiplexing recipient;  
and

an instance of said event message deliverable from said multiplexing re-  
cipient to at least one intended recipient of said event message.

## ABSTRACT OF THE DISCLOSURE

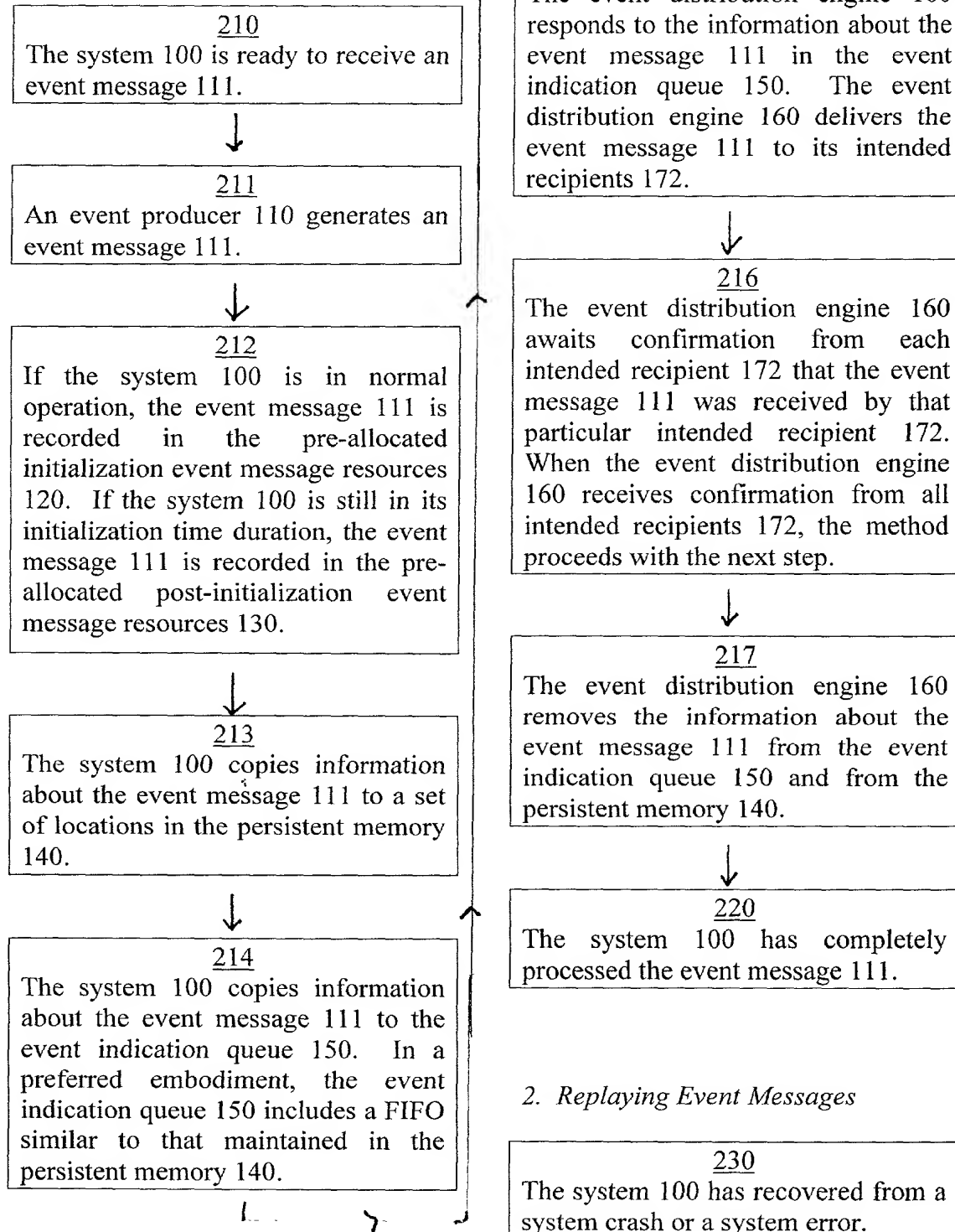
The invention provides a method and system for persistent and reliable delivery of event messages. Those parts of the system responsible for delivering event messages are able to persistently maintain those event messages until the intended recipient of the event message confirms delivery of those event messages. Those parts of the system responsible for recovering from system crashes and other system errors are able to persistently maintain those event messages until delivery, even after recovery from system crashes or other system errors. The system includes a set of event message producers, and maintains an event-indication queue of those event messages provided by the event producers using a set of pre-allocated resources. An event-distribution engine distributes event messages to intended recipients and, after having received confirmation that the event messages were received, removes them from the event-indication queue. Recipients of event messages receive the event messages, acknowledge their receipt thereof, and might take action in response to the event message. The system includes persistent memory, initialization memory, and recipient persistent memories, and provides upon recovery from system crashes or other system error, an ability to replay event messages recorded in those memories, to re-present them as event messages. A cluster of file servers collectively forming a highly-available system shares persistent memories, so that upon a system crash or other system error, at least one other file server has a record of those event messages.

003180-1048-01





### 1. Processing Event Messages



### 2. Replaying Event Messages

to Fig 2B

From Fig 2A

231

The event replay engine 190 reads information about event messages 111 from the persistent memory 140.



231(a)

The system initialization replay sub-element 191 reads information about event messages 111 associated with the pre-allocated post-initialization event message resources 130. The event replay engine 190 replays these event messages 111.



231(b)

The incomplete event distribution replay sub-element 192 reads information about event messages 111 associated with the pre-allocated initialization event message resources 120. The event replay engine 190 replays these event messages 111 next.



231(c)

The cooperating systems replay sub-element 193 reads information about event messages 111, from the persistent memory 140, associated with and stored there by a cooperating system 100.



240

The system 100 has replayed all event messages 111 not yet fully processed, and is ready to proceed at the flow point 210.

### 3. Multiplexing Recipient Operation

250

A multiplexing recipient 171 is ready to receive an event message 111.



251

The multiplexing recipient 171 receives the event message 111 from the event distribution engine 160.



252

The multiplexing recipient 171 records information about the event message 111 in its second persistent memory 180.



253

The multiplexing recipient 171 (optionally) responds to the event message 111 by confirming that it was received at the multiplexing recipient 171 (but not necessarily at the intended recipients 172).



254

The multiplexing recipient 171 (optionally) determines to which intended recipients 172 to deliver the event message 111.



255

The multiplexing recipient 171 awaits confirmation from each particular intended recipient 172 that the particular intended recipient 172 has received the event message 111.



to Fig 2C

From 2B



256

The multiplexing recipient 171 receives such confirmation from individual intended recipients 172.



260

The multiplexing recipient 171 has completely processed the event message 111.

#### 4. *Replaying Multiplexed Event Messages*

270

The multiplexing recipient 171 has recovered from a system crash or a system error.



271

The multiplexing recipient 171 reads information about event messages 111 from the second persistent memory 180.



272

The multiplexing recipient 171 replays the event messages 111 from the second persistent memory 180.



280

The multiplexing recipient 171 has replayed all event messages 111 not yet fully processed, and is ready to proceed at the flow point 250.

#### 5. *Confirming Event Messages*

290

The intended recipient 172 is ready to receive an event message 111.



291

The intended recipient 172 receives an event message 111.



292

The intended recipient 172 parses the event message 111 and processes the event message 111 according to its own (internal) processing rules for that event message 111.



293

The intended recipient 172 generates a confirmation message and sends that confirmation message to the sender of the event message 111.



300

The intended recipient 172 has received, processed, and confirmed the event message 111. The sender of the event message 111, upon receipt of the confirmation message, can regard the event message 111 as completely handled and can safely delete it.